Hands-On Session

Workflow for this Hands-On Session

1. Write Python code to solve the programming questions.
2. All the questions are based on what we have already discussed.
3. Solutions to all the questions are available in the Day3_Solutions_1.ipynb Jupyter notebook.
4. It is highly encouraged to solve the programming questionnaires on your own even though you can refer to the solutions.

Q1. Use the get method to print the value of the "model" key of the car dictionary.

In [ ]:

```
car =   {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
print(car.get("model"))
```

Q2. Write a python code to loop through any given dictionary.

In [1]:

```
mydict = {'StuName': 'Steve', 'StuAge': 4, 'StuCity': 'Agra'}
for e in mydict:
  print("Key:",e,"Value:",mydict[e])
```

```
Key: StuName Value: Steve
Key: StuAge Value: 4
Key: StuCity Value: Agra
```

Q3. Write a Python script to generate and print a dictionary that contains a number (between 1 and n) in the form (x, x*x)

In [11]:

```
n=int(input("Input a number "))
d = dict()

for x in range(1,n+1):
    d[x]=x*x

print(d)
```

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}
```

Q4. Create a dictionary by extracting the key values from a given dictionary. The given dictionary is sample_dict{"name": "Kelly", "age": 25, "salary": 8000, "city": "New york"} and extract the name and salary from the given dictionary

In [2]:

```
sample_dict = {
    "name": "Kelly",
    "age": 25,
    "salary": 8000,
    "city": "New york"
}
# Keys to remove
keys = ["name", "salary"]

for k in keys:
    sample_dict.pop(k)
print(sample_dict)
```

```
{'age': 25, 'city': 'New york'}
```

Q5. Write a Python script to store book id, book name and price of three books and store the data in dictionary named "dict"

In [9]:

```
dict = {}
n = int(input("Enter the number of books:"))
for i in range(n):
     b_id = int(input("Enter book id : "))
     b_name = input("Enter book name : ")
     b_price = int(input("Enter price of book : "))
     temp=[b_name,b_price]
     dict[b_id] = temp
print(dict)
```

```
{33: ['Python', 444]}
```

Q6. Write a Python script to generate a Dictionary that contains cities located in India and their corresponding population. Calculate the sum of the population and the average population Delhi – 8550405, Bangalore – 3971883, Chennai – 2731571, Hyderabad – 2720546, Jaipur – 2296224, Udaipur – 1704694

In [5]:

```
city_population = {"Delhi": 8550405,"Bangalore": 3971883,
                   "Chennai": 2731571,"Hyderabad": 2720546,
                   "Jaipur": 2296224,"Udaipur": 1704694}
print(city_population)
average = (sum(city_population.values()) / len(city_population))
print(f"Average of all the population is {average}")
```

```
{'Delhi': 8550405, 'Bangalore': 3971883, 'Chennai': 2731571, 'Hyderabad': 27
20546, 'Jaipur': 2296224, 'Udaipur': 1704694}
Average of all the population is 3662553.8333333335
```

Q7. Given a dictionary, append another dictionary at beginning of it. Append the {'pre1': 4, 'Gfg': 5} to the beginning of the test_dict={"Gfg":5, "is":3, "best":10}

In [3]:

```
# Python3 code to demonstrate working of
# Append items at beginning of dictionary
# Using update()

# initializing dictionary
test_dict = {"Gfg" : 5, "is" : 3, "best" : 10}

# printing original dictionary
print("The original dictionary is : " + str(test_dict))

# initializing update dictionary
updict = {"pre1" : 4, "pre2" : 8}

# update() on new dictionary to get desired order
updict.update(test_dict)

# printing result
print("The required dictionary : " + str(updict))
```

```
The original dictionary is : {'Gfg': 5, 'is': 3, 'best': 10}
The required dictionary : {'pre1': 4, 'pre2': 8, 'Gfg': 5, 'is': 3, 'best':
10}
```

Q8. Write a Python program to filter the height and width of students, which are stored in a dictionary. {'Omkar': (6.2, 70), 'Sohan': (5.9, 65), 'Yatin': (6.0, 68), 'Yash': (5.8, 66)} Find the student with Height > 6ft and Weight> 70kg: {'Omkar': (6.2, 70)} .

In [4]:

```
def filter_data(students):
    result = {k: s for k, s in students.items() if s[0] >=6.0 and s[1] >=70}
    return result

students = {'Omkar': (6.2, 70), 'Sohan': (5.9, 65), 'Yatin': (6.0, 68), 'Yash': (5.8, 66)}
print("Original Dictionary:")
print(students)
print("\nHeight > 6ft and Weight> 70kg:")
print(filter_data(students))
```

```
Original Dictionary:
{'Omkar': (6.2, 70), 'Sohan': (5.9, 65), 'Yatin': (6.0, 68), 'Yash': (5.8, 6
6)}

Height > 6ft and Weight> 70kg:
{'Omkar': (6.2, 70)}
```

Q9. Write a Python program to remove a specified dictionary from a given list. [{'id': '#FF0000', 'color': 'Red'}, {'id': '#800000', 'color': 'Maroon'}, {'id': '#FFFF00', 'color': 'Yellow'}, {'id': '#808000', 'color': 'Olive'}] Remove id #FF0000 from the said list of dictionary: [{'id': '#800000', 'color': 'Maroon'}, {'id': '#FFFF00', 'color': 'Yellow'}, {'id': '#808000', 'color': 'Olive'}]

In [5]:

```
def remove_dictionary(colors, r_id):
    colors[:] = [d for d in colors if d.get('id') != r_id]
    return colors

colors = [{"id" : "#FF0000", "color" : "Red"},
          {"id" : "#800000", "color" : "Maroon"},
          {"id" : "#FFFF00", "color" : "Yellow"},
          {"id" : "#808000", "color" : "Olive"}]
print('Original list of dictionary:')
print(colors)
r_id = "#FF0000"
print("\nRemove id",r_id,"from the said list of dictionary:")
print(remove_dictionary(colors, r_id))
```

```
Original list of dictionary:
[{'id': '#FF0000', 'color': 'Red'}, {'id': '#800000', 'color': 'Maroon'},
{'id': '#FFFF00', 'color': 'Yellow'}, {'id': '#808000', 'color': 'Olive'}]

Remove id #FF0000 from the said list of dictionary:
[{'id': '#800000', 'color': 'Maroon'}, {'id': '#FFFF00', 'color': 'Yellow'},
{'id': '#808000', 'color': 'Olive'}]
```

Sample programs on Tuples

Type *Markdown* and LaTeX: α^2

In [3]:

```
# Different types of tuples

# Empty tuple
my_tuple = ()
print(my_tuple)

# Tuple having integers
my_tuple = (1, 2, 3)
print(my_tuple)

# tuple with mixed datatypes
my_tuple = (1, "Hello", 3.4)
print(my_tuple)

# nested tuple
my_tuple = ("mouse", [8, 4, 6], (1, 2, 3))
print(my_tuple)
```

```
()
(1, 2, 3)
(1, 'Hello', 3.4)
('mouse', [8, 4, 6], (1, 2, 3))
```

Q11. Write a python program to concatenate two tuples

In [4]:

```
# Concatenation of tuples
Tuple1 = (0, 1, 2, 3)
Tuple2 = ('Python', 'For', 'Beginners')

Tuple3 = Tuple1 + Tuple2

# Printing first Tuple
print("Tuple 1: ")
print(Tuple1)

# Printing Second Tuple
print("\nTuple2: ")
print(Tuple2)

# Printing Final Tuple
print("\nTuples after Concatenation: ")
print(Tuple3)
```

```
Tuple 1:
(0, 1, 2, 3)

Tuple2:
('Python', 'For', 'Beginners')

Tuples after Concatenation:
(0, 1, 2, 3, 'Python', 'For', 'Beginners')
```

Q12. Write a Python program to get the 4th element and 4th element from last of a tuple

In [8]:

```
#Get an item of the tuple
tuplex = ("p", "y", "t", "h", "o", "n", " ", "P", "r", "o","g", "r", "a", "m")
print("The original tuple is ",tuplex)
#Get item (4th element)of the tuple by index
item = tuplex[3]
print(item)
#Get item (4th element from last)by index negative
item1 = tuplex[-4]
print(item1)
```

```
The original tuple is  ('p', 'y', 't', 'h', 'o', 'n', ' ', 'P', 'r', 'o',
'g', 'r', 'a', 'm')
h
g
```

Q13. Write a Python program to reverse the contents of a tuple.

In [9]:

```
#create a tuple
x = ("Python Programming")
# Reversed the tuple
y = reversed(x)
print(tuple(y))
#create another tuple
x = (5, 10, 15, 20)
# Reversed the tuple
y = reversed(x)
print(tuple(y))
```

```
('g', 'n', 'i', 'm', 'm', 'a', 'r', 'g', 'o', 'r', 'P', ' ', 'n', 'o', 'h',
't', 'y', 'P')
(20, 15, 10, 5)
```

Q14. Write a Python program to test if a variable is a list or tuple or a set.

In [10]:

```
#x = ['a', 'b', 'c', 'd']
#x = {'a', 'b', 'c', 'd'}
x = ('tuple', False, 3.2, 1)
if type(x) is list:
    print('x is a list')
elif type(x) is set:
    print('x is a set')
elif type(x) is tuple:
    print('x is a tuple')
else:
    print('Neither a list or a set or a tuple.')
```

```
x is a tuple
```

Q15. Write a python program to create a nested tuple to store roll number, name and marks of students

In [12]:

```
st=((101,"Aman",98),(102,"Geet",95),(103,"Sahil",87),(104,"Pawan",79))
print("S_No"," Roll_No"," Name"," Marks")
for i in range(0,len(st)):
    print((i+1),'\t',st[i][0],'\t',st[i][1],'\t',st[i][2])
```

```
S_No  Roll_No  Name  Marks
1        101     Aman    98
2        102     Geet    95
3        103     Sahil   87
4        104     Pawan   79
```

Q16.Write a program to input n numbers from the user. Store these numbers in a tuple. Print the maximum and minimum number from this tuple.

In [16]:

```
numbers = tuple() #create an empty tuple 'numbers'
n = int(input("How many numbers you want to enter?: "))
for i in range(0,n):
    num = int(input())
 #it will assign numbers entered by user to tuple 'numbers'
    numbers = numbers +(num,)
print('\nThe numbers in the tuple are:')
print(numbers)
print("\nThe maximum number is:")
print(max(numbers))
print("The minimum number is:")
print(min(numbers))
```

```
The numbers in the tuple are:
(11, 99, 33, 77, 66)

The maximum number is:
99
The minimum number is:
11
```

Q17. Write the Python program to sort the elements of the given tuple

In [17]:

```
my_tuple = (12, 2, 5, 90, 30, 40, 3)
print(f'The tuple is - {my_tuple}')
result_tuple = sorted(my_tuple)
print(f'Increasing order of the tuple is - {result_tuple}')
result_tuple = sorted(my_tuple, reverse=True)
print(f'Decreasing order of the tuple is - {result_tuple}')
```

```
The tuple is - (12, 2, 5, 90, 30, 40, 3)
Increasing order of the tuple is - [2, 3, 5, 12, 30, 40, 90]
Decreasing order of the tuple is - [90, 40, 30, 12, 5, 3, 2]
```

Python File Handling Programs

Q18. Write a Python program to create a text file “intro.txt” in python and ask the user to write a single line of text by user input.

In [18]:

```
def program1():
    f = open("intro.txt","w")
    text=input("Enter the text:")
    f.write(text)
    f.close()
program1()
```

Q19. Write a python program to create a text file “MyFile.txt” in python and ask the user to write separate 3 lines with three input statements from the user.

In [19]:

```
def program2():
    f = open("MyFile.txt","w")
    line1=input("Enter the text:")
    line2=input("Enter the text:")
    line3=input("Enter the text:")
    new_line="\n"
    f.write(line1)
    f.write(new_line)
    f.write(line2)
    f.write(new_line)
    f.write(line3)
    f.write(new_line)
    f.close()
program2()
```

Q20. Write a python program to read the contents of both the files created in the above programs and merge the contents into “merge.txt”. Avoid using the close() function to close the files.

In [20]:

```
def program3():
    with open("MyFile.txt","r") as f1:
       data=f1.read()
    with open("intro.txt","r") as f2:
        data1=f2.read()
    with open("merge.txt","w") as f3:
        f3.write(data)
        f3.write(data1)
program3()
```

Q21. Write a python program to count the total number of upper case, lower case, and digits used in the text file “merge.txt”.

In [21]:

```
def program4():
    with open("merge.txt","r") as f1:
       data=f1.read()
    cnt_ucase =0
    cnt_lcase=0
    cnt_digits=0
    for ch in data:
        if ch.islower():
            cnt_lcase+=1
        if ch.isupper():
            cnt_ucase+=1
        if ch.isdigit():
            cnt_digits+=1
    print("Total Number of Upper Case letters are:",cnt_ucase)
    print("Total Number of Lower Case letters are:",cnt_lcase)
    print("Total Number of  digits are:",cnt_digits)
program4()
```

```
Total Number of Upper Case letters are: 6
Total Number of Lower Case letters are: 29
Total Number of  digits are: 3
```

Q22. Write a python program to count a total number of lines and count the total number of lines starting with 'A', 'B', and 'C'. (Consider the merge.txt file)

In [24]:

```
def program5():
    with open("merge.txt","r") as f1:
       data=f1.readlines()
    cnt_lines=0
    cnt_A=0
    cnt_B=0
    cnt_C=0
    for lines in data:
        cnt_lines+=1
        if lines[0]=='A':
            cnt_A+=1
        if lines[0]=='B':
            cnt_B+=1
        if lines[0]=='C':
            cnt_C+=1
    print("Total Number of lines are:",cnt_lines)
    print("Total Number of lines strating with A are:",cnt_A)
    print("Total Number of lines strating with B are:",cnt_B)
    print("Total Number of lines strating with C are:",cnt_C)
program5()
```

```
Total Number of lines are: 4
Total Number of lines strating with A are: 0
Total Number of lines strating with B are: 0
Total Number of lines strating with C are: 0
```

Q23. write a python program to find the total occurrences of a specific word from a text file:

In [25]:

```
def program6():
    cnt = 0
    word_search = input("Enter the words to search:")
    with open("merge.txt","r") as f1:
        for data in f1:
            words = data.split()
            for word in words:
                if (word == word_search):
                    cnt+=1
    print(word_search, "found ", cnt, " times from the file")
program6()
```

```
Veena found  1  times from the file
```

Q24. Write a python program to read first n no. letters from a text file, read the first line, read a specific line from a text file.

In [28]:

```
def program7():
    cnt = 0
    n = int(input("Enter no. characters to read:"))
    with open("merge.txt","r") as f1:
       line1=f1.readline()
       print("The first line of file:",line1)
       nchar=f1.read(n)
       print("First n no. of characters:", nchar)
       nline=f1.readlines()
       print("Line n:",nline[n])
program7()
```

```
The first line of file: Program1

First n no. of characters: Pr
Line n: My name is Veena A
```

Q25. Write a python program to replace all spaces from text file with a – (dash).

In [30]:

```
def program8():
    cnt = 0
    n = int(input("Enter no. characters to read:"))
    with open("merge.txt","r") as f1:
       data = f1.read()
       data=data.replace(' ','-')
    with open("merge.txt","w") as f1:
        f1.write(data)
program8()
```

Q26. Write a program to know the cursor position and print the text according to below-given specifications: 1.Print the initial position 2.Move the cursor to 4th position 3.Display next 5 characters 4.Move the cursor to the next 10 characters 5.Print the current cursor position 6.Print next 10 characters from the current cursor position.

In [31]:

```
def program9():
    f = open("merge.txt","r")
    print(f.tell())
    f.seek(4,0)
    print(f.read(5))
    f.seek(10,0)
    print(f.tell())
    print(f.seek(7,0))
    print(f.read(10))
program9()
```

```
0
ram1

10
7
1
Program2
```

Q27. Write a python program to append the contents in entered by the user in the text file:

In [32]:

```
def program10():
    text = input("Enter text to append in the file:")
    with open("merge.txt","a") as f1:
        f1.write(text)
program10()
```

Q28. Write a python program to read the contents of file in reverse order

In [33]:

```
def program11():
    for i in reversed(list(open("merge.txt","r"))):
        print(i.rstrip())
program11()
```

```
My-name-is-Veena-AWelcome to the World of Programming
Program3
Program2
Program1
```

Q29. Write a python program to replace multiple spaces with single space in a text file.

In [35]:

```
def program12():
    f1 = open("merge.txt","rt")
    f2 = open("merge1.txt","wt")
    for line in f1:
        f2.write(' '.join(line.split()))
    f1.close()
    f2.close()
program12()
```

In [ ]:

```

```